WEIGHTED FEATURE SELECTION CRITERIA FOR VISUAL SERVOING OF A TELEROBOT

John T. Feddema, C. S. G. Lee, and O. R. Mitchell

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

Because of the continually changing environment of a space station, visual feedback is a vital element of a telerobotic system. A real-time visual servoing system would allow a telerobot to track and manipulate randomly moving objects. This paper develops methodologies for the automatic selection of image features to be used to visually control the relative position between an eye-in-hand telerobot and a known object. A weighted criteria function with both image recognition and control components is used to select the combination of image features which provides the best control. Simulation and experimental results of a PUMA robot arm visually tracking a randomly moving carburetor gasket with a visual update time of 70 milliseconds are discussed.

I. Introduction

Most would agree that the eventual goal of a telerobot is to perform dangerous tasks in space which would otherwise require human intervention. To perform these tasks, telerobots must be equipped with many of the sensory capabilities of humans. Because of the continually changing environment of a space station, vision is undoubtably a very important sense. Until recently, the primary uses of vision in telerobotics have been for recognizing, locating, and inspecting stationary parts. Image processing equipment is now reaching the stage where vision may be used as a feedback signal to control the position and orientation (pose) of the telerobot's end-effector in real time [1][2]. This visual feedback would allow a telerobot to manipulate and track a randomly moving part without any previous knowledge of the part's placement or motion.

The type of feedback for visual servoing systems has taken two forms [3]: position-based and feature-based. The traditional method has been to extract image features †, recognize the desired object by matching image features to a set of pre-taught features, interpret the pose of the object based on the image features, and use the error between desired and estimated pose to drive the system. A second method is to use actual image features instead of the part's interpreted position as the feedback signal for controlling the manipulator [4]. With the proper selection, these features can be directly related to the control parameters of the robotic system. The savings in time needed to interpret the workpiece's pose from the image features is made possible by determining the desired image features during an off-line CAD simulation.

In this paper, a resolved motion rate control scheme [5] with feature-based feedback are used to visually servo a robot manipulator with an eye-in-hand camera over a moving object (see Figure 1). This method of visual feedback introduces two fundamental questions. How many image features are necessary to control the desired degrees of freedom of the manipulator end-effector? And which image features would provide the best control? This paper addresses these two questions and develops methodologies for the automatic selection of image features used to visually control the relative pose between the manipulator end-effector and a workpiece. The selection of these features depends on a blend of image recognition and control criteria. The image recognition criteria include feature robustness, completeness, uniqueness, and cost of feature extraction. The control criteria include system observability, controllability, and sensitivity. A weighted criteria function is used to select the combination of image features which provides the best control. Both computer simulations and laboratory experiments on a PUMA robot arm were conducted to verify the performance of the feature selection criteria.

This work was supported in part by an IBM research fellowship and in part by the National Science Foundation under Grant CDR 8803017 to the Engineering Research Center for Intelligent Manufacturing Systems. Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the funding agencies.

†In this paper, image features refer to geometric shapes in the camera's image. Examples include circles, edges, corners, and curves.

II. Differential Relationship between Part's Pose and Image Feature Points

In the resolved motion rate control structure in Figure 1, the changes in image features are transformed into changes in joint angles. This transformation may be decomposed into a series of three transformations: feature to camera coordinates, camera to end-effector coordinates, and end-effector to robot joint coordinates. The last two transformations are well known [6] but the first is not. For our purposes, we will assume that a unique transformation from camera space to robot joint space exists. Therefore, our ability to control the pose of the robot depends on the differential transformation from image feature space to camera space. This section concentrates on the differential transformation from image feature points †† to camera coordinates.

When analyzing the transformations from the camera space to the image feature space, consider the coordinate frames shown in Figure 2. The following nomenclature is used:

(x , y , z) = position of the part with respect to the camera frame;

(ϕ , θ , ψ) = roll, pitch, and yaw orientation of the part with respect to the camera frame;

$({}^p x_i , {}^p y_i , {}^p p_i)$ = position of feature point i on the part with respect to the part frame;

$({}^c x_i , {}^c y_i , {}^c z_i)$ = position of feature point i on the part with respect to the camera frame;

$({}^I x_i , {}^I y_i)$ = corresponding position of the point in the image plane;

f = focal length of the camera lens;

γ_x = x axis scaling factor in pixels/mm due to camera sampling;

γ_y = y axis scaling factor in pixels/mm due to camera sampling; and

(x_o , y_o) = the image plane offset in pixels due to camera sampling.

We assume that the camera characteristics $(f, \gamma_x, \gamma_y, x_o, y_o)$ are known. To be able to interpret the 3-D pose of a part, we have assumed that the spatial relationships between feature points, i.e., $({}^p x_i , {}^p y_i , {}^p z_i)$, are known from a CAD model.

Geometric optics are used to model the mapping between the Cartesian space and the image feature space. The mapping consists of two stages: a thin lens model of the perspective transformation, and a mapping into a two-dimensional plane caused by camera sampling. The transformation from the camera frame, c, to the image plane, I, can be written as

$$\begin{bmatrix} {}^I x_i w_i \\ {}^I y_i w_i \\ w_i \end{bmatrix} = \begin{bmatrix} \gamma_x & 0 & 0 & x_o \\ 0 & \gamma_y & 0 & y_o \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & f \\ 0 & 0 & -1 & f \end{bmatrix} \begin{bmatrix} {}^c x_i \\ {}^c y_i \\ {}^c z_i \\ 1 \end{bmatrix} \tag{1}$$

or

$${}^I x_i = \gamma_x \left[\frac{f}{f - {}^c z_i} \right] {}^c x_i + x_o \qquad \text{and} \qquad {}^I y_i = \gamma_y \left[\frac{f}{f - {}^c z_i} \right] {}^c y_i + y_o \tag{2}$$

where we assume that the blurring effects, the quantization, and the lens distortion effects are negligible.

Using the differential transform technique [6], the change in image positions $(d^I x_i, d^I y_i)$ in terms of the change in the part's pose $(d_x, d_y, d_z, \delta_x, \delta_y, \delta_z)$ is [7]

$$\begin{bmatrix} d^I x_i \\ d^I y_i \end{bmatrix} = \begin{bmatrix} J_{x_i d_x} & J_{x_i d_y} & J_{x_i d_z} & J_{x_i \delta_x} & J_{x_i \delta_y} & J_{x_i \delta_z} \\ J_{y_i d_x} & J_{y_i d_y} & J_{y_i d_z} & J_{y_i \delta_x} & J_{y_i \delta_y} & J_{y_i \delta_z} \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \\ \delta_x \\ \delta_y \\ \delta_z \end{bmatrix} \tag{3}$$

where

$$J_{x_i d_x} = \frac{\gamma_x f}{f - {}^c z_i} , \qquad J_{x_i d_y} = 0 , \qquad J_{x_i d_z} = \frac{\gamma_x f \, {}^c x_i}{(f - {}^c z_i)^2} ,$$

†† The (x , y) position of a feature such as a circle or corner in the image will be referred to as a feature point.

$$J_{x_i\delta_x} = \frac{\gamma_x f\,{}^c x_i\,{}^c y_i}{(f - {}^c z_i)^2}\,, \qquad J_{x_i\delta_y} = \frac{\gamma_x f}{f - {}^c z_i}\left[{}^c z_i - \frac{{}^c x_i^2}{f - {}^c z_i}\right], \qquad J_{x_i\delta_z} = \frac{-\gamma_x f\,{}^c y_i}{f - {}^c z_i}\,,$$

$$J_{y_i d_x} = 0\,, \qquad J_{y_i d_y} = \frac{\gamma_y f}{f - {}^c z_i}\,, \qquad J_{y_i d_z} = \frac{\gamma_y f\,{}^c y_i}{(f - {}^c z_i)^2}\,,$$

$$J_{y_i\delta_x} = \frac{\gamma_y f}{f - {}^c z_i}\left[-{}^c z_i + \frac{{}^c y_i^2}{f - {}^c z_i}\right], \qquad J_{y_i\delta_y} = \frac{-\gamma_y f\,{}^c x_i\,{}^c y_i}{(f - {}^c z_i)^2}\,, \quad \text{and} \quad J_{y_i\delta_z} = \frac{\gamma_y f\,{}^c x_i}{f - {}^c z_i}\,.$$

This expression is suitable for simulation purposes where we want to determine the change in image features given a small change in the part's pose. However, for control purposes we seek an inverse solution. We would like to know the change in the part's pose given the change in several feature points in the image. An exact inverse may exist if three feature points are considered. The linear differential relationship between the change in image feature points and the change in the part's pose would then be 6 × 6 matrix **J**.

Notice that this Jacobian matrix depends on the positions of the feature points with respect to the camera. For the inverse Jacobian matrix **J** to exist, all three points can not have the same x and y position in the image plane. If the points are collinear, then a rotation about the line would not be observable. Another restriction is that not all feature points can be on a plane perpendicular to the focal axis with one of the points located at the focal center. Again, there exists a rotation in this plane which the camera will not be able to sense.

One final problem still remains. How can we use this transformation if we do not know the actual positions of the feature points with respect to the camera? In our experiments, two approaches were used. The first was to estimate their positions from their image positions and the known spatial relationships between image points. The second was to use the desired positions as determined by the CAD simulation. While the latter is self-explanatory, the first method deserves further development.

The objective is to find the pose of the part with respect to the camera, i.e., $(x, y, z, \phi, \theta, \psi)$, from the a priori information, $({}^p x_i\,, {}^p y_i\,, {}^p z_i)$, and the measured feature points in the image, $({}^I x_i\,, {}^I y_i)$, for $i = 1, 2, \cdots, n$, where n is the number of feature points in the image. Variations of this "location determination problem" have appeared in several papers [8]. Overdetermined systems of equations are typically used to determine the "best" solution. Least squares techniques, the Random Sample Consensus paradigm [8], and a generalized Hough transform approach [9] have been proposed to eliminate errors caused by noisy images and modeling errors.

Unfortunately, many of these methods are too time consuming for real-time use. Instead, we propose using the following gradient search to continually update the object's pose. This search minimizes the sum of squared distance errors between four or more actual feature points and the geometrically modeled feature points in the image. Because of the nature of this search, a fairly good initial estimate of the part's pose is necessary. It should also be noted that the resulting least squares solution is optimal if the image noise has a Gaussian distribution. However, if there are outliers in the data, the Random Sample Consensus approach [8] might provide more accurate results.

The objective of this gradient search is to vary the pose of the workpiece until the image positions of the modeled feature points align with the actual image feature points. In mathematical terms, we would like to minimize

$$F = \sum_{i=1}^{n} F_i = \frac{1}{2}\sum_{i=1}^{n} ({}^I x_i - u_i)^2 + ({}^I y_i - v_i)^2 \tag{4}$$

where $(u_i\,, v_i)$ is the actual image position of feature point i. The modeled position of feature point i in the image is determined from the camera's perspective and sampling Eq. (2), and from the roll, pitch, yaw representation of the part with respect to the camera [6],

$$\begin{bmatrix} {}^c x_i \\ {}^c y_i \\ {}^c z_i \\ 1 \end{bmatrix} = \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi & x \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi & y \\ -s\theta & c\theta s\psi & c\theta c\psi & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^p x_i \\ {}^p y_i \\ {}^p z_i \\ 1 \end{bmatrix}, \tag{5}$$

where $c\phi \equiv cos(\phi)$, $s\phi \equiv sin(\phi)$, $c\theta \equiv cos(\theta)$, $s\theta \equiv sin(\theta)$, $c\psi \equiv cos(\psi)$, and $s\psi \equiv sin(\psi)$.

In these equations, the known variables are the camera parameters, $(f, \gamma_x, \gamma_y, x_o, y_o)$; the actual image feature positions, (u_i, v_i), for $i = 1, 2, \cdots, n$; and the positions of the features with respect to the part's frame, $({}^p x_i\,, {}^p y_i\,, {}^p z_i)$, for $i = 1, 2, \cdots, n$, where n is the number of features. The unknown variables (the ones we want to solve for) are the pose of the part, $(x, y, z, \phi, \theta, \psi)$. For ease of notation, let $\mathbf{x} = [x, y, z, \phi, \theta, \psi]^T$, $\mathbf{y}_i = [{}^c x_i\,, {}^c y_i\,, {}^c z_i]^T$, and $\mathbf{z}_i = [{}^I x_i\,, {}^I y_i]^T$, where the superscript T denotes vector/matrix transpose. Then Eq. (2) can be represented by $\mathbf{z}_i = \mathbf{G}(\mathbf{y}_i)$, and Eq. (5) can be represented by $\mathbf{y}_i = \mathbf{H}_i(\mathbf{x})$.

Newton's gradient search is used to minimize this error function with respect to the $\mathbf{x}$ parameters. The kth iteration of the search is given by

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{F}_{\mathbf{xx}}^{-1}(\mathbf{x}_k)\ \nabla F^T(\mathbf{x}_k) \tag{6}$$

where ∇F is the gradient of F with respect to x and the $\mathbf{F}_{\mathbf{xx}}$ is the Hessian matrix of F with respect to $\mathbf{x}$.

Because the relationship between the error function and the part's pose consists of a series of composite mappings described in Eqs. (2), (4), and (5), the gradient of F is the product of the Jacobian matrices of individual mappings. The gradient of F with respect to x is

$$\nabla F = \sum_{i=1}^{n} \mathbf{F}_{i\mathbf{z}_i}\ \mathbf{G}_{\mathbf{y}_i}\ \mathbf{H}_{i\mathbf{x}} \tag{7}$$

where $\mathbf{F}_{i\mathbf{z}_i}$ is the Jacobian matrix of F_i with respect to $\mathbf{z}_i$, $\mathbf{G}_{\mathbf{y}_i}$ is the Jacobian matrix of $\mathbf{G}$ with respect to $\mathbf{y}_i$, and $\mathbf{H}_{i\mathbf{x}}$ is the Jacobian matrix of $\mathbf{H}_i$ with respect to $\mathbf{x}$. The elements of these Jacobian matrices are given in [7].

The chain rule for the Hessian matrix $\mathbf{F}_{xx}$ is slightly more difficult.

$$\mathbf{F}_{\mathbf{xx}} = \sum_{i=1}^{n} \mathbf{H}_{i\mathbf{x}}^T\ \mathbf{G}_{\mathbf{y}_i}^T\ \mathbf{F}_{i\mathbf{z}_i\mathbf{z}_i}\ \mathbf{G}_{\mathbf{y}_i}\ \mathbf{H}_{i\mathbf{x}} + \frac{\partial F_i}{\partial^I x_i}\frac{\partial^2({}^I x_i)}{\partial \mathbf{x}^2} + \frac{\partial F_i}{\partial^I y_i}\frac{\partial^2({}^I y_i)}{\partial \mathbf{x}^2} \tag{8}$$

where the Hessian matrix of ${}^I x_i$ with respect to $\mathbf{x}$ is

$$\frac{\partial^2({}^I x_i)}{\partial \mathbf{x}^2} = \mathbf{H}_{i\mathbf{x}}^T\ \frac{\partial^2({}^I x_i)}{\partial \mathbf{y}_i^2}\ \mathbf{H}_{i\mathbf{x}} + \frac{\partial^I x_i}{\partial^c x_i}\ \frac{\partial^2({}^c x_i)}{\partial \mathbf{x}^2} + \frac{\partial^I x_i}{\partial^c y_i}\ \frac{\partial^2({}^c y_i)}{\partial \mathbf{x}^2} + \frac{\partial^I x_i}{\partial^c z_i}\ \frac{\partial^2({}^c z_i)}{\partial \mathbf{x}^2}\ , \tag{9}$$

and the Hessian matrix of ${}^I y_i$ with respect to $\mathbf{x}$ is

$$\frac{\partial^2({}^I y_i)}{\partial \mathbf{x}^2} = \mathbf{H}_{i\mathbf{x}}^T\ \frac{\partial^2({}^I y_i)}{\partial \mathbf{y}_i^2}\ \mathbf{H}_{i\mathbf{x}} + \frac{\partial^I y_i}{\partial^c x_i}\ \frac{\partial^2({}^c x_i)}{\partial \mathbf{x}^2} + \frac{\partial^I y_i}{\partial^c y_i}\ \frac{\partial^2({}^c y_i)}{\partial \mathbf{x}^2} + \frac{\partial^I y_i}{\partial^c z_i}\ \frac{\partial^2({}^c z_i)}{\partial \mathbf{x}^2}\ . \tag{10}$$

The elements of these Hessian matrices are also in [7].

III. Feature Selection Criteria

Since the Jacobian obtained in the previous section depends on the features' positions in the 3-D space with respect to the camera, some features will provide better visual control than others as the part moves with respect to the camera. In addition to these control issues, image recognition plays an important role in choosing features which are reliable and robust. This section lists several image recognition and control criteria which should be considered in the selection process. This is not an all inclusive list and additional criteria could be added at a latter date.

The image recognition criteria used in our feature selection process include [10]:

1. rare features (similar to unique features in [11]),
2. feature set robustness (similar to likelihood of being seen in [11]),
3. computational inexpensive features, and
4. feature set completeness.

To quantitatively evaluate these criteria, a set of measures was designed for the visual servoing experiments. For uniformity, these measures were designed to range between zero and one with zero being the most desirable and one being the least desirable.

A unique feature has an easily identifiable characteristic which differs from other features in the image. Such a feature is often used to quickly identify an object. This makes the feature very useful for control since the feature can be quickly re-identified if it is momentarily lost. A measure of feature uniqueness for a set of features $\{f_1, \cdots, f_m\}$ may be written as

$$\chi_1(f_1, \cdots, f_m) = \frac{1}{Nm(n-1)}\left[\sum_{i=1}^{m}\sum_{j=1}^{N} M_{ij}\right] \tag{11}$$

where m is the number of features to be used for control, n is the total number of features in the images, N is the number of possible views during the control process, and M_{ij} is the number of similar features to feature i in image j $(0 \le M_{ij} \le n-1)$. Note that the measure χ_1 reaches a maximum of one if $M_{ij} = n-1$ for all i and j and a minimum of zero if $M_{ij} = 0$ for all i and j.

In order that the recognition process be resilient to noise, we would like the features to be robust. For feature extraction systems where the image is scanned for features, feature robustness depends on the size and type of feature and the method of feature extraction. Usually the larger the feature is in the image, the more robust the feature is. A general measure of feature robustness may be written as:

$$\chi_2(f_1\,,\,\cdots\,,f_m)=\frac{1}{Nm}\left[\sum_{i=1}^{m}\sum_{j=1}^{N}(1-\sigma_{ij})\,\rho_{ij}\right] \tag{12}$$

where $0 \le \sigma_{ij} \le 1$ is proportional to the size of the feature i in the image j, and $0 \le \rho_{ij} \le 1$ is related to the reliability of feature extraction method ij. In our experiments, features were restricted to circles in the image. Since the same method of feature extraction was used throughout the experiments, the reliability factor is $\rho_{ij} = 1$ for all i,j. Our measure of feature robustness for circles was

$$\chi_2(f_1\,,\,\cdots\,,f_m)=1-\frac{1}{Nmr_{\max}}\left[\sum_{i=1}^{m}\sum_{j=1}^{N}r_{ij}\right] \tag{12.a}$$

where r_{ij} was the radius of circle j in image j, and $r_{\max}$ was the maximum radius possible.

The computational expense of features refers to the time and space complexities of the feature extraction process. For most cases, space complexity is negligible. Time complexity, on the other hand, is very important for determining the tracking ability of the visual servoing control. The shorter the time of feature extraction is, the larger the bandwidth of the control. For feature extraction systems where the image is scanned for features, computational expense also depends on the size and type of feature and the method of feature extraction. In contrast to the robustness criteria, usually the smaller the feature is in the image, the smaller the computational expense is. A general measure of computational expense of features may be written as:

$$\chi_3(f_1\,,\,\cdots\,,f_m)=\frac{1}{Nm}\left[\sum_{i=1}^{m}\sum_{j=1}^{N}\sigma_{ij}\,\tau_{ij}\right] \tag{13}$$

where $0 \le \tau_{ij} \le 1$ is related to the time complexity of the feature extraction method ij for a feature of fixed size. In our experiments, the location of a circle was verified with four scan lines spanning out from the approximate origin of the circle to the circle's edge. Again, since the same method of feature extraction was used throughout the experiments, $\tau_{ij} = 1$ for all i,j. Our measure of computational expense for the circles was

$$\chi_3(f_1\,,\,\cdots\,,f_m)=\frac{1}{Nmr_{\max}}\left[\sum_{i=1}^{m}\sum_{j=1}^{N}r_{ij}\right]\ . \tag{13.a}$$

Notice the trade-off between the computational expense and robustness criteria. Smaller circles have less computational cost with poor feature robustness, while larger circles have better robustness at the expense of computational cost.

If a workpiece can be identified from any view point, the set of features is said to be complete. For our circumstances where we are continually controlling the position of the workpiece with respect to the camera, only a subset of all possible views is necessary. Our measure for feature set completeness is

$$\chi_4(f_1\,,\,\cdots\,,f_m)=\frac{1}{Nm}\left[\sum_{i=1}^{m}\sum_{j=1}^{N}u_{ij}\right]\qquad where\qquad u_{ij}=\begin{cases}0\,, & \text{if feature } i \text{ is in image } j\\ 1\,, & \text{if feature } i \text{ is not in image } j\,.\end{cases} \tag{14}$$

The control criteria used in our feature selection process include:

1. Observability of the workpiece's pose through image feature points.
2. Controllability of the workpiece's pose using the inverse Jacobian matrix in section II.
3. Sensitivity of the control to noise.

In the presense of image noise, the gradient search in section II was used to minimize the error in the image positions of four or more features. Since this error is measurable, it provides a means of evaluating the observability of the part's pose with respect to the camera. Using the error function F in Eq. (22), our measure for the observability of the part's pose is

$$\chi_5(f_1\,,\,\cdots\,,f_m)=\frac{1}{NF_{\max}}\sum_{j=1}^{N}\hat{F}\qquad where\qquad \hat{F}=\begin{cases}F\,, & \text{if } F \le F_{\max}\\ F_{\max}\,, & \text{if } F > F_{\max}\,,\end{cases} \tag{15}$$

and $F_{\max}$ is the largest acceptable error.

If the sampling time between image acquisitions is short and the distance between the actual and desired image features is small, the differential relationship in section II could be used to control the relative pose of the workpiece. The desired change in camera position would be determined by multiplying the difference between actual and desired features by the inverse Jacobian. In this respect, we will say that the pose of the workpiece is "controllable" if the inverse Jacobian exists and is non-singular.

In addition to checking that a unique solution exists, we would like the equations to be well-conditioned. This means that for "small" changes in **J** and the part's pose, the changes in image feature positions should also be small. This could be thought of a measure of the control's sensitivity to noise. The condition of the Jacobian matrix **J** is

$$c(\mathbf{J}) = ||\,\mathbf{J}\,||\ ||\,\mathbf{J}^{-1}\,|| \tag{16}$$

where the norm may be $||\cdot||_1$, $||\cdot||_2$, or $||\cdot||_\infty$. Moderately small values of $c(\mathbf{J})$ imply that the equations are well-conditioned. However, large values of $c(\mathbf{J})$ do not necessarily mean that the equations are ill-conditioned. Instead, it just means that the equations will be ill-conditioned for some changes in the part's pose. In general, we would like to choose image feature points which would minimize the condition of **J**. Since the condition of **J** will become extremely large as the Jacobian approaches a singular point, the condition may also be used to evaluate the controllability of the workpiece's pose using the inverse Jacobian matrix. Therefore, a measure for evaluating both the controllability and sensitivity of the system may be written as

$$\chi_6(f_1, f_2, f_3) = \frac{1}{Nc_{\max}} \sum_{j=1}^{N} \hat{c}(\mathbf{J}) \qquad \textit{where} \qquad \hat{c}(\mathbf{J}) = \begin{cases} c(\mathbf{J}), & \text{if } \mathbf{J}^{-1} \text{ exists and } c(\mathbf{J}) \le c_{\max}, \\ c_{\max}, & \text{if } \mathbf{J}^{-1} \text{ exists and } c(\mathbf{J}) > c_{\max}, \\ c_{\max}, & \text{if } \mathbf{J}^{-1} \text{ does not exist}, \end{cases} \tag{17}$$

and $c_{\max}$ is the largest acceptable value for the condition of **J**.

Another consideration for real-time control is the effect that changes in the feature positions have on the elements of the Jacobian matrix. To reduce computational costs, it is desirable for the elements of the Jacobian to be constant or slowly time-varying. If the elements do not change substantially between two camera acquisitions, the inverse Jacobian does not have to be updated each sample time. Therefore, we would like to choose image feature points such that the elements of the Jacobian change very little throughout the motion. In other words, we would like to minimize the sensitivity of **J** to the change of image feature points,

$$s(\mathbf{J}) = \sum_{i=1}^{3} \sum_{k=0}^{6} \sum_{m=0}^{6} \left\{ \left| \frac{\partial \mathbf{J}_{km}}{\partial\, {}^{c}x_i} \cdot \frac{{}^{c}x_i}{\mathbf{J}_{km}} \right| + \left| \frac{\partial \mathbf{J}_{km}}{\partial\, {}^{c}y_i} \cdot \frac{{}^{c}y_i}{\mathbf{J}_{km}} \right| + \left| \frac{\partial \mathbf{J}_{km}}{\partial\, {}^{c}z_i} \cdot \frac{{}^{c}z_i}{\mathbf{J}_{km}} \right| \right\} \tag{18}$$

where $\mathbf{J}_{km}$ are the elements of the Jacobian. Our measure of sensitivity to change was

$$\chi_7(f_1, f_2, f_3) = \frac{1}{Ns_{\max}} \sum_{j=1}^{N} \hat{s}(\mathbf{J}) \qquad \textit{where} \qquad \hat{s}(\mathbf{J}) = \begin{cases} s(\mathbf{J}), & \text{if } s(\mathbf{J}) \le s_{\max}, \\ s_{\max}, & \text{if } s(\mathbf{J}) > s_{\max}, \end{cases} \tag{19}$$

and $s_{\max}$ is the largest acceptable value of $s(\mathbf{J})$.

Because of the various conflicting interests of the above measures, a weighted criteria function was used to select the features which would provide the best control

$$\chi(f_1, \cdots, f_m) = \sum_{i=1}^{7} w_i\, \chi_i(f_1, \cdots, f_m) \tag{20}$$

where $\sum_{i=1}^{7} w_i = 1$ are the weighting factors which are the choice of the designer. Similar to the individual measures, the best selection will be the set of features with the smallest overall measure χ.

IV. Simulation and Experimental Results

Both computer simulations and laboratory experiments were performed to illustrate the performance of the proposed feature selection criteria. Computer simulations were used to test the performance of the feature selection criteria under ideal conditions and with additive image noise. The experiments verified the simulation results and showed the usefulness of the feature selection criteria for visually controlling a robot manipulator in real time.

In our experiments, a single Pulnix TM-540 CCD camera mounted on the end-effector of a PUMA 600 robot was used to visually servo the robot's end-effector over a moving carburetor gasket (the workpiece) as shown in Figure 3. Because of their ease of feature extraction, the circles in the gasket were used as the control features. The visual servo control was initiated when the camera was approximately over the gasket in a pre-taught position. After the gasket was recognized, the locations of three circles in the image were continually updated. The changes in camera pose needed to control the robot were determined by multiplying the difference between the desired and actual image positions by the inverse Jacobian in section II. The changes in camera pose were then converted to the changes in joint angles using the manipulator's inverse Jacobian [12].

The simulations were modeled after the experiments and their objective was to determine which three image feature points (circles) of the carburetor gasket out of a total of seven circles (see Figure 3) would be best suited for controlling the pose of the gasket with respect to the camera. Table 1 lists the positions with respect to the gasket's frame and the radii of the seven largest holes. The number of feature points was limited to seven to reduce the number of possible combinations. The desired position and orientation of the gasket with respect to the camera was (x, y, z) = (-48,63,200) millimeters and (roll, pitch, yaw) = (-90,0,0) degrees, respectively. The camera parameters are given in Table 2.

First, we evaluated the condition and sensitivity of **J** for all possible combinations of three circles. We found that the set of features which minimize $c(\mathbf{J})$ form an equilateral triangle about the origin of the gasket. In particular, the set of circles {0,3,5} minimized $c(\mathbf{J})$ while the set of circles {0,1,2} maximized $c(\mathbf{J})$. We also found that the condition of **J** decreases as the gasket is moved closer to the camera. On the other hand, the sensitivity of **J** to change was minimized as the feature points moved closer to the focal axis and the x and y image axes. According to this sensitivity criteria, the best set of features would be the set of circles {1,2,4}. In contrast to the condition of **J**, the sensitivity of **J** to change is minimized if the gasket is as far away from the camera as possible. If computation time was a factor, a weighted sum of the measures χ_6 and χ_7 could be used to determine a set of features which would provide accurate control without updating the inverse Jacobian.

Next, we considered the tracking response of the camera for the two sets of features, {0,3,5} and {0,1,2}. For both sets of features, circle 6 was used as a fourth point for determining location. To test the control process, a ramp input in position and orientation was applied to the gasket's pose. The change in the x, y, and z directions was 5 millimeters per sample. The change in the roll, pitch, and yaw of the gasket was 0.9 degrees per sample. If our system was operating at video rate (30 Hz - both fields), this would correspond to a positional velocity of 15 centimeters per second and a rotational velocity of 27 degrees per second. Several cases were run with and without image noise.

First consider the ideal case with no image noise. Table 3 shows the root mean square (RMS) error between the desired and actual position and orientation of the workpiece with respect to the camera. Three schemes were used to update the Jacobian matrix. In column 1, the Jacobian was updated with the actual positions of the feature points. In column 2, the Jacobian was updated only once with the desired positions of the feature points. In column 3, the Jacobian was updated with the estimated positions of the feature points using the methods in section II. The following conclusions were made from the ideal case:

1. Updating the Jacobian with the actual positions provides the best control.
2. When the actual or desired positions of the features were used, the set of features {0,3,5} performed as well as the feature set {0,1,2}.
3. As predicted by the large condition of the Jacobian for feature set {0,1,2}, it was difficult to estimate the part's pose from the feature set {0,1,2}. Therefore, for estimation purposes it is best to choose a set of feature points with a small condition number, such as the feature set {0,3,5}.

Next consider the same simulation except add Gaussian image noise with a standard deviation of 0.5 pixels. The RMS error for this case is shown in Table 4. In addition to the same columns as in Table 3, a fourth column shows the results when seven points are used to estimate the pose of the workpiece. The columns with ∞ indicate that the workpiece was lost before the simulation was completed. The following conclusions can be made from the noisy image case:

1. Updating the Jacobian with the desired positions provides the best control.
2. Even when using desired or actual positions to update the Jacobian, image noise has a costly effect on the resolved motion rate control of a set of features with a small condition number, such as the feature set {0,1,2}. Therefore, for control purposes it is best to choose a set of feature points with a small condition number, such as the feature set {0,3,5}.
3. The more points used to estimate the part's pose, the better the estimate will be.

In the experiments, the camera was first moved to the desired position over the gasket and the selection criteria were evaluated as shown in Table 5. The criteria weighting factors were set to $\{w_1, w_2, w_3, w_4, w_5, w_6, w_7\}$ = {0.1, 0.15, 0.05, 0.1, 0.2, 0.2, 0.2}. The computation time of feature extraction was weighted less than the other criteria since there was not a noticeable difference in delay time when extracting one circle over the next. The maximum acceptable pose estimation error, condition, and sensitivity were F_{max} = 100, c_{max} = 100,000, and s_{max} = 68.

After the features were selected, the gasket was placed on a conveyor belt moving at a speed of 2.7 centimeters per second in the y and z directions of the world coordinates. Using the resolved motion rate visual feedback and a feature-based trajectory generator [12], the PUMA robot tracked the gasket. The feature-based trajectory generator had a smoothing effect on image noise by spreading out the change in image features over time. To speed up real-time computation and avoid estimation noise effects the desired positions of the feature points were used to compute the Jacobian. This works satisfactorily if the feature points are already in the vicinity of the desired positions.

In Figure 4, the first 37 seconds show the ramp response of the system along the z direction while the last 10 seconds show the steady-state error when the conveyor belt was stopped. The response in the y direction was a similar ramp shaped graph. The responses in the x direction and the orientation angles were small oscillatory signals about the desired values. The maximum positional errors in the x, y, and z directions were 18.4, 21.6, and 8.0 millimeters, respectively. The maximum orientation errors in roll, pitch, and yaw were 1.5, 2.6, and 2.2 degrees, respectively. The oscillation in the x position and the orientation angles can be attributed to image noise and the feedback delay time. In order to increase the vision sampling, the verification process used to update the position of the circles was very simple. Unfortunately, it was also fairly noisy. During the servoing process, the circles in the image were located with an accuracy of ± 5 pixels. The delay time between when the image was taken and when the robot actuation began was approximately 100 milliseconds.

V. Conclusion

Methodologies for the automatic selection of image features used to visually control the relative pose between the manipulator end-effector and a workpiece were developed and analyzed. A resolved motion rate control scheme was used to update the robot's pose based on the position of three features in the camera's image. The selection of these three features for control was based on a weighted criteria function with both image recognition and control components. Because of the real-time nature of the control process, it is important to find reliable image features which can be quickly extracted from the image. Of particular importance for resolved motion rate control was the condition of the Jacobian matrix relating the differential change in the workpiece's pose to the corresponding differential change in the image feature points. To minimize the effects of image noise on the control, the condition of the Jacobian should be minimized. In the simulations and experiments, we found that estimating the pose of the workpiece was a time consuming and noise sensitive process. Because of this, the best tracking results were obtained by using the desired feature locations to compute the Jacobian instead of the estimated locations.

VI. References

[1] B. Wilcox, et al. "The Sensing and Perception Subsystem of NASA Research Telerobot," *Proceeding of the Workshop on Space Telerobotics*, G. Rodriguez, editor, JPL Publ. 87-13, July 1, 1987.

[2] P. Symosek, et al. "Knowledge-Based Vision for Space Station Object Motion Detection, Recognition, and Tracking," *Proceeding of the Workshop on Space Telerobotics*, JPL Publ. 87-13, July 1, 1987.

[3] A.C. Sanderson and L.E. Weiss, "Adaptive Visual Servo Control of Robots," Reprinted in *Robot Vision*, Alan Pugh, editor, IFS Publications Ltd., 1983.

[4] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic Sensor-Based Control of Robots with Visual Feedback," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 5, pp. 404-417, October 1987.

[5] D.E. Whitney, "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 122, pp. 303-309, December 1972.

[6] R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*, The MIT Press, Cambridge, Massachusetts, 1981.

[7] J.T. Feddema, C.S.G. Lee, and O.R. Mitchell, "Optimal Selection of Image Features for Resolved Rate Visual Feedback Control," Purdue University, Technical Report TR-ERC 89-3, January 1989.

[8] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. of the ACM*, Vol. 24, No. 6, pp. 381-395, June 1981.

[9] S. Linnainmaa, D. Harwood, and L.S. Davis, "Pose Determination of a Three-Dimensional Object Using Triangle Pairs," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 5, pp. 634-647, Sept. 1988.

[10] C. Hansen and T. Henderson, "CAGD-Based Computer Vision," *IEEE Workshop on Computer Vision*, pp. 100-105, November 1987. Ellis Horwood Publishers, New York, pp. 81-84, 1985.

[11] R. Horaud and T. Skordas, "Model-Based Strategy Planning for Recognizing Partially Occluded Parts," *Computer*, pp. 58-64, August 1987.

[12] J.T. Feddema and O.R. Mitchell, "Vision Guided Servoing with Feature-Based Trajectory Generation," *SME World Conference on Robotic Research*, Gaithersburg, Maryland, May 9-12, 1989.

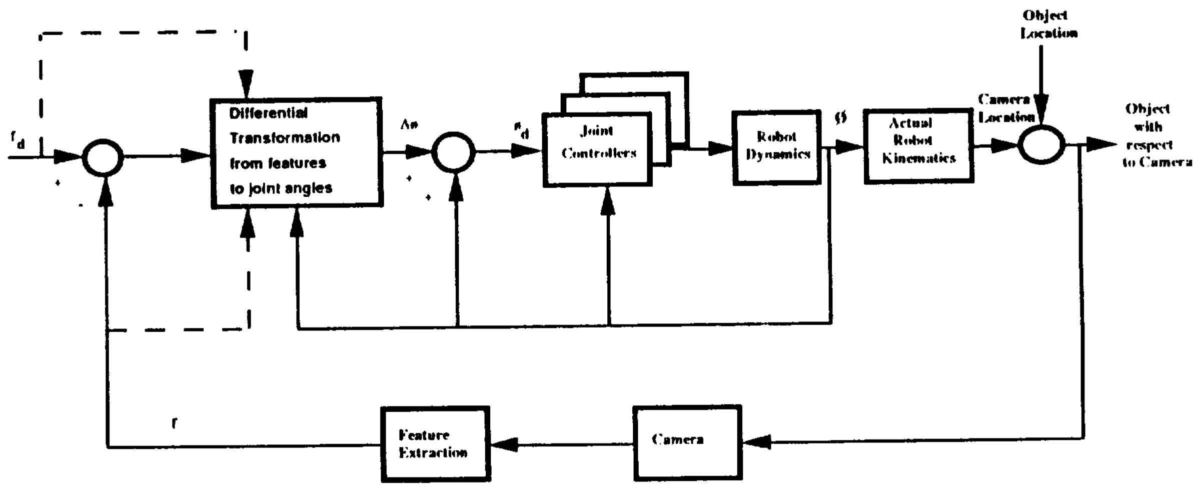

Figure 1. Resolved motion rate control structure for visual feedback.

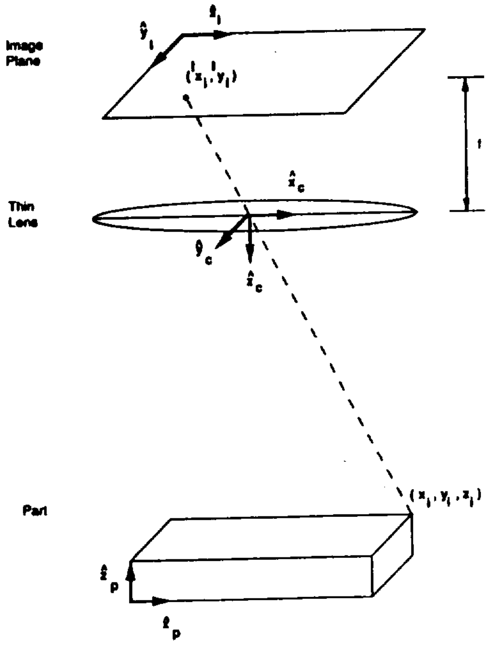

Figure 2. Part, camera, and image coordinate frames.

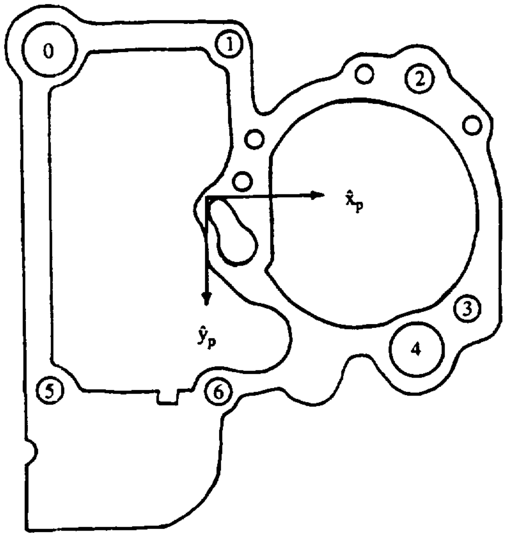

Figure 3. Seven image feature points of the carburetor gasket.

Table 1. Positions and radii of the circles with respect to the gasket's frame.

Positions and radii in millimeters				
No.	x	y	z	radius
0	98	5	0	5
1	98	42	0	3
2	93	80	0	3
3	46	91	0	3
4	37	81	0	5
5	28	6	0	3
6	28	39	0	3

Table 2. Intrinsic parameters of the camera.

Parameter	Symbol	Value
Focal length	f	8.0 mm
X scale factor	γ_x	-67.2832 pixels/mm
Y scale factor	γ_y	-84.7279 pixels/mm
X focal center	x_o	240.0 pixels
Y focal center	y_o	240.0 pixels

Table 3. Ideal case: RMS error of gasket with respect to the camera.

Control Features	Experiment		#1	#2	#3
	Jacobian Update		Actual	Desired	Estimated *
{0,3,5}	RMS Estimation Error	Position (mm)	0	0	0.000034
		Orientation (deg)	0	0	0.000032
	RMS Control Error	Position (mm)	0.2847	0.3526	0.2847
		Orientation (deg)	0.1493	0.1722	0.1493
{0,1,2}	RMS Estimation Error	Position (mm)	0	0	7.6440
		Orientation (deg)	0	0	4.5745
	RMS Control Error	Position (mm)	0.1987	0.1798	0.9727
		Orientation (deg)	0.0360	0.0430	0.2940

* Jacobian updated only if least squares search error F is small.

Table 4. Image noise case: RMS error of gasket with respect to the camera. The image noise had a Normal distribution with zero mean and a standard deviation of 0.5 pixels. The table shows the mean position and orientation RMS error of 10 trials.

Control Features	Experiment		#1	#2	#3	#4
	Jacobian Update		Actual	Desired	Estimate #1 *	Estimate #2 **
{0,3,5}	RMS Estimation Error	Position (mm)	0	0	2.5663	2.2569
		Orientation (deg)	0	0	2.7393	2.1801
	RMS Control Error	Position (mm)	1.2708	1.1542	1.5735	1.4578
		Orientation (deg)	1.3395	1.2593	1.5454	1.4490
{0,1,2}	RMS Estimation Error	Position (mm)	0	0	∞	∞
		Orientation (deg)	0	0	∞	∞
	RMS Control Error	Position (mm)	∞	∞	∞	∞
		Orientation (deg)	∞	∞	∞	∞

* Estimate #1 implies that four circles were used to estimate the gasket's pose.

** Estimate #2 implies that seven circles were used to estimate the gasket's pose.

Table 5. The best combinations of features with weights {0.1, 0.15, 0.05, 0.1, 0.2, 0.2, 0.2}.

Feature Selection Table For Experiments											
Circles				Selection Criteria							
#1	#2	#3	#4	χ_1	χ_2	χ_3	χ_4	χ_5	χ_6	χ_7	χ
0	3	5	4	0.341	0.219	0.781	0.000	0.023	0.110	0.919	0.316
0	4	5	3	0.341	0.219	0.781	0.000	0.033	0.148	0.916	0.325
0	2	4	3	0.341	0.219	0.781	0.000	0.041	0.139	0.919	0.326
0	3	6	4	0.341	0.216	0.784	0.000	0.013	0.171	0.917	0.326
0	4	5	6	0.341	0.237	0.762	0.000	0.047	0.133	0.916	0.327
0	2	3	4	0.341	0.219	0.781	0.000	0.042	0.159	0.921	0.330
1	4	5	0	0.341	0.235	0.765	0.000	0.042	0.178	0.915	0.335
0	4	6	3	0.341	0.216	0.784	0.000	0.024	0.220	0.915	0.338
0	2	6	4	0.341	0.238	0.762	0.000	0.106	0.130	0.917	0.338
0	4	5	2	0.341	0.241	0.759	0.000	0.119	0.117	0.916	0.339

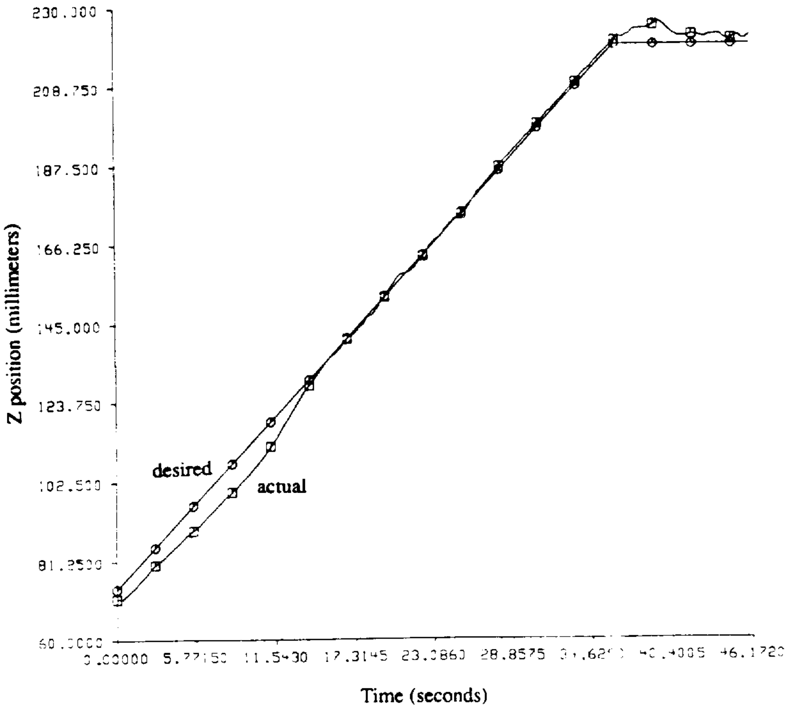

Figure 4. Robot end-effector's z position while tracking the gasket in the experiments.